



MINECRAFT

EDUCATION EDITION

Educator Guide

Single Student

Coding Concept: VARIABLES

Variables are used to store information to be referenced and manipulated in a computer program. They also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves. It is helpful to think of variables as containers that hold information. Their sole purpose is to label and store data in memory. This data can then be used throughout your program.

Naming variables is known as one of the most difficult tasks in computer programming. When you are naming variables, think hard about the names. Try your best to make sure that the name you assign your variable is accurately descriptive and understandable to another reader. For more information on **variables**, see: <https://curriculum.code.org/csd-18/unit3/5/>

Variables in Minecraft:

Interacting with **variables** in Minecraft: Education Edition can make it easier for the player to build more effectively, control the outcome of code and even capture input from players for later use. Players can create their own variables, define the parameters associated with them and then use the newly created variables in their code to execute complex instructions for the agent. Any player who understands the use of variables has the advantage over players that do not.

OVERVIEW

These learning activities are designed to provide the learner with flexibility and choice in his/her learning.

1. The learner will be introduced to three tutorials that are at three levels of difficulty (Novice, Intermediate and Expert) which are followed by an educational activity where he/she can apply coding skills.
2. Within each tutorial, the learner will have the option of selecting a coding language (Blocks, JavaScript or Python).

The learner may choose a tutorial at his/her appropriate difficulty level and language OR choose to complete all tutorials and the educational activity. It will take approximately an hour to complete all activities.

LESSON STRUCTURE

Section 1: Mega Jump

Section 2: Bat Cave

Section 3: Letter Printing

Education Application:

The learner is free to engage with activities in Block coding, JavaScript and/or Python and does not need to complete all the activities.

The learner will visit several different locations where they will assess the available resources and then collect items to build a **sustainable** dwelling. There are many elements of sustainability to consider when building a structure in the real world, including available natural resources, climate, building space, etc.

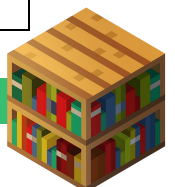
LEARNING OBJECTIVES





The learner will:

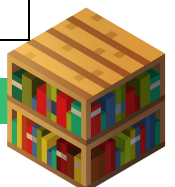
- Learn to collect appropriate resources that are **sustainable** within a given biome.
- Learn to manipulate **variables** in the provided code to control/direct the agent.
- Learn how to use the **agent** to produce a structure that is **sustainable** within a given **biome**.


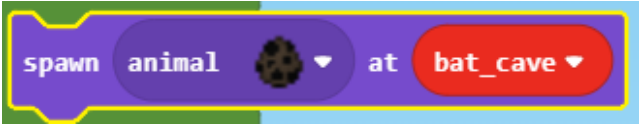
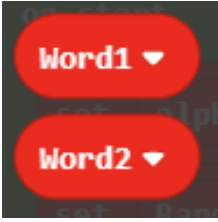

CODING BLOCKS

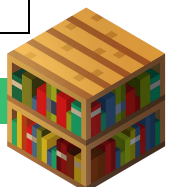
Definition	Code
on chat command Runs the code when the student types the	Blocks:




Definition	Code
<p>chosen text in the chat window.</p>	 <p>JavaScript: <pre>player.onChat("", function () {</pre> <pre>}}</pre> Python: <pre>def on_chat():</pre> <pre> pass</pre> <pre>player.on_chat("", on_chat)</pre> </p>
<p>teleport to Runs code that transports your player to the coordinates provided.</p>	<p>Blocks:</p>  <p>JavaScript: <pre>player.onChat("jump", function () {</pre> <pre> player.teleport(pos(0,100,0))</pre> <pre>})</pre> Python: <pre>def on_chat():</pre> <pre> player.teleport(pos(0, 100, 0))</pre> <pre>player.on_chat("jump", on_chat)</pre> </p>
<p>repeat Runs the code a defined number of times.</p>	<p>Blocks:</p>  <p>JavaScript: <pre>for (let index = 0; index < 50; index++) {</pre> <pre>)</pre> <pre>}</pre> Python: <pre>for index in range(50):</pre> </p>
<p>set variable to Allows the player to define the location where a variable is used.</p>	<p>Blocks:</p>  <p>JavaScript: <pre>batcave = player.position()</pre> </p>



Definition	Code
	<p>Python: batcave = player.position()</p>
<p>say Displays text on the screen for the player to see.</p>	<p>Blocks:</p>  <p>JavaScript: player.say("Watch out for bats!")</p> <p>Python: player.say("Watch out for bats!")</p>
<p>spawn animal at Generates an animal into Minecraft at a defined position.</p>	<p>Blocks:</p>  <p>JavaScript: mobs.spawn(BAT, batcave)</p> <p>Python: mobs.spawn(BAT, batcave)</p>
<p>variable Used to define parameters that hold values, numbers or characters.</p>	<p>Blocks:</p>  <p>JavaScript: Let starting_world_position: Position = null let word1 = "" word1 = "rail"</p> <p>Python: word1 = "rail" word2 = "road"</p>
<p>join Used to connect variables.</p>	<p>Blocks:</p>  <p>JavaScript: "" + word1 + word2,</p> <p>Python: "" + word1 + word2</p>



Definition	Code
<p>print Used to produce an output in blocks defined by its parameters.</p>	<p>Blocks:</p>  <p>JavaScript: blocks.print()</p> <p>Python: blocks.print ()</p>

KEYWORDS (OR PHRASES)

1. **Agent** - Your assistant who helps you do things in Minecraft. You command the agent with the agent blocks. Your agent appears when you teleport it to you. When you have your agent, it can do work for you.
2. **Command** is an instruction defined by the player to produce an intended result when executed, can be simple or complex.
3. **World Position** is three numbers that are the distances in each direction from the world origin: **(0, 0, 0)**. This is also called an *absolute* position.
4. **Relative Position** has ~ before each number, is centered on the player's current position: (~0, ~0, ~0), starts from the player's feet, is not fixed in the world.
5. **Spawn** means to bring a creature into the game at a position you choose.
6. **Set** is an assignment operator that uses an equals sign to make a variable store the number, string, or Boolean. When you use the equals sign to store something in a variable, the equals sign is called an *assignment operator*, and what you store is called a *value*.
7. **Say** posts a message to the game chat where other players can read it and react accordingly.

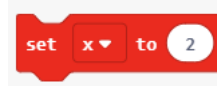
GUIDING QUESTIONS

1. Can you define **sustainability**?
 - a. *The ability to exist constantly.*
 - b. *In the 21st century, it refers generally to the capacity for the biosphere and human civilization to coexist.*
 - c. *Sustainability focuses on meeting the needs of the present without compromising the ability of future generations to meet their needs.*
2. Can you give an example of **sustainability**?
 - a. *The Rain Forest is disappearing at an alarming rate which is having a dramatic impact on the rivers due to the large amounts of the soil now being washed into the rivers, killing fish and other aquatic life which has a direct impact on those who depend on the fish for their food source, including people.*
 - b. *As populations increase and cities expand to try to keep pace, farmland is lost, food production decreases and the increasing population runs the risk of running out of food.*



- c. *Building a home in the desert out of blocks of ice would not be sustainable however, the same ice block home in the Arctic would be easily sustainable.*
- 3. Can you define the term **variable** in coding (computer science)? Can you give an example of a **variable** in coding (Computer Science)?
 - a. *A variable represents the place in memory for the data that's kept there. It's called a variable because the data it represents may change (vary) when the program stores something new there.*
 - b. *Examples include, x, y, z, count, index, looping.*
- 4. How can/does a player use **variables** in Minecraft? Can you give an example?
 - a. *Use the Block Editor variable statement to create a variable and the assignment operator to store something in the variable. This is called declaring the variable.*

b. `let percent = 50;`



- 5. How can using **variables** help your game play in Minecraft?
 - a. *Executing repetitive actions in Minecraft is time consuming. Using variables in combination with loops allows the player to execute complex instructions with much less player interaction, working more efficiently and accomplishing more in less time. Let the agent build your walls while you tend to the chickens.*

TUTORIAL 1: MEGA JUMP (Novice-10 minutes)

Main objective:

The learner will utilize the **teleport** command and coordinates to cause their player to jump great distances.

Answer Key: <https://makecode.com/TMAVm5Tk44JT>

TUTORIAL 2: BAT CAVE (Intermediate-15 minutes)

Main objective:

The learner will utilize **repeat** and **fill** commands to create an underground space and then **spawn** bats into that space.

Answer Key: <https://makecode.com/Cx5evTgXa7y8>

TUTORIAL 3: LETTER PRINTING (Expert-20 minutes)

Main objective:

The learner will create and implement several **variables** to produce block printed letters and words at desired locations.

Answer Key: <https://makecode.com/M9TDzDR830b8>

EDUCATIONAL APPLICATION CASE (25 minutes)

Subject Area: Science

The learner will visit several different locations where they will assess the available resources and then collect items to use in a final build.

The learner's challenge is to determine which items to collect so that when they return to build site, they will be able to supply the provided code with the necessary **variables** with which to build a sustainable dwelling.



The world has four unique locations and a *Start* and *Finish* location that are the same. The other four locations have several items available (4-5) each, but only 1 or 2 are useful at the build site.

- Learner will first evaluate the starting location based on weather/biome conditions/available resources, topography, etc.
- Learner will then visit each remaining location, one at a time, collecting resources for a build at the final location.
- Learner will return to the build site and use a “starter code” that includes an **Agent** or **Builder** with the code to build a simple structure (cabin) out of the materials they picked up along the way.

The concept of **Sustainability** will be supported by introducing **variables** as parameters for the “structure” they build at their starting/finishing location as they use the provided build code but modify the:

- NUMBER of blocks for the “length”, “width”, “height” of the structure (based on how many they collected on their journey)
- “Type” of blocks used in the structure (only from the ones they collected & contributes to the design of the structure itself)
- Coordinates of the build site (x, y, z) such that the structure actually fits (is physically sustainable on the site)

Outcome of this activity:

The learner will have successfully made decisions about what type of blocks to pick up, how many, and where to place them on the build site. They do this by manipulating the stated variables (length, width, height, type and (x, y, z) coordinates) in the provided code. The resulting build will be determined by the decisions they make, which is a core component of the concept of **Sustainability**.

Possible Solution for this Activity: attached

LESSON CONCLUSION/REFLECTION (10 minutes)

Ask the learners about the skills that they have practiced during the lesson to reinforce the concepts.

1. Name some of the **variables** used in this lesson.
(length, width, height, type and (x, y, z) coordinates)
2. What is a **variable** in coding/Computer Science?
***Variables** are used to store information to be referenced and manipulated in a computer program. They also provide a way of labeling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves.*
3. What is **Sustainability**? (as it relates to natural resources, weather, economy, etc.)
Sustainability is about making decisions with regard to natural resources, weather, economy, etc. that enable us to continue to exist over time without destroying our local environment (planet).
4. In what way were the **variables** (length, width, height, type and (x, y, z) coordinates) in this activity constrained (affected) by **sustainability**?
The limited number of blocks determined how large the structure could be. This limited the (length, width, height). The topography of the build site determined where it was safe to build (x, y, z) coordinates. The (type) of blocks collected directly affected the design of the final build.



OPPORTUNITIES TO EXTEND THE LEARNING

Minecraft: Education Edition offers many opportunities for teachers and learners to extend and enrich the learning experience beyond this lesson.

In this activity students learned how to use numbers and blocks as variables in code that was given to them that the agent could execute for them.

- Learners could extend the use of variables in this or any world/activity by first writing their own code for the agent and introducing new variables to their own code as defined by the learner.
- Students could work through the variables lesson from Code.org here: <https://curriculum.code.org/csd-18/unit3/5/>

EDUCATION STANDARDS

NEXT GENERATION SCIENCE STANDARDS (NGSS) (<https://www.nextgenscience.org/>)

HS-ESS3-3 Earth and Human Activity: Create a computational simulation to illustrate the relationships among management of natural resources, the sustainability of human populations, and biodiversity.

COMPUTER SCIENCE TEACHERS ASSOCIATION (CSTA) COMPUTER SCIENCE STANDARDS K-12

<https://www.csteachers.org/Page/standards>

1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to represent information.

1B-AP-09: Create programs that use variables to store and modify data.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

INTERNATIONAL SOCIETY FOR TECHNOLOGY IN EDUCATION (ISTE) STANDARDS FOR STUDENTS

<https://www.iste.org/standards/for-students>

1C: Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

4A: Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artefacts or solving authentic problems.

5A: Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.

